IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 24, 2020, accepted July 13, 2020, date of publication July 23, 2020, date of current version August 5, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3011590

Detection of Stock Price Manipulation Using Kernel Based Principal Component Analysis and Multivariate Density Estimation

BAQAR A RIZVI, (Member, IEEE), AMMAR BELATRECHE, (Member, IEEE), AHMED BOURIDANE, (Senior Member, IEEE), AND IAN WATSON
Department of Computer and Information Sciences, Northumbria University, Newcastle upon Tyne NE1 8ST, U.K.

Corresponding author: Ammar Belatreche (ammar.belatreche@northumbria.ac.uk)

This work was supported by the Research and Development Fund, Department of Computer and Information Sciences, Northumbria University, Newcastle upon Tyne, U.K.

ABSTRACT Stock price manipulation uses illegitimate means to artificially influence market prices of several stocks. It causes massive losses and undermines investors' confidence and the integrity of the stock market. Several existing research works focused on detecting a specific manipulation scheme using supervised learning but lacks the adaptive capability to capture different manipulative strategies. This begets the assumption of model parameter values specific to the underlying manipulation scheme. In addition, supervised learning requires the use of labelled data which is difficult to acquire due to confidentiality and the proprietary nature of trading data. The proposed research establishes a detection model based on unsupervised learning using Kernel Principal Component Analysis (KPCA) and applied increased variance of selected latent features in higher dimensions. A proposed Multidimensional Kernel Density Estimation (MKDE) clustering is then applied upon the selected components to identify abnormal patterns of manipulation in data. This research has an advantage over the existing methods in overcoming the ambiguity of assuming values of several parameters, reducing the high dimensions obtained from conventional KPCA and thereby reducing computational complexity. The robustness of the detection model has also been evaluated when two or more manipulative activities occur within a short duration of each other and by varying the window length of the dataset fed to the model. Validation on multiple datasets and a comprehensive assessment of the model performance has been conducted without providing any prior information about the location of the manipulation. The results show a significant performance enhancement in terms of the F-measure values and a significant reduction in false alarm rate (FAR) has been achieved.

INDEX TERMS Market abuse, stock price manipulation, anomaly detection, kernel principal component analyses, multi-dimensional kernel density estimate clustering.

I. INTRODUCTION

Stock market manipulation creates a false impression of stock prices through some illegitimate means [1]. It not only affects investor's interest in the manipulated stocks but also undermines their confidence in the integrity of the entire market. Allen and Gale [2] classified market manipulation into three main types: action based, information based, and trade based manipulation. Action based manipulation is an action rather than trading, performed by the company managers or executives who hold the supply of a well-established product by increasing its demand and hence the stock price. Information based manipulation intends to spread a false rumor or release some inside information about a company or its stock with an intention to influence the price. Trade based manipulation on the other hand has everything to do inside a stock exchange where traders, investors or brokers buy/sell stocks at different prices for different volumes [2], [3]. One of the major types of trade-based manipulation is price manipulation in which the trader targets to influence the buy/sell prices of any company stock. It should also be noted that this type of manipulation is excessively used and hence has the largest impact on stock markets [3]. Additionally, unlike the first two types of manipulation that can be avoided

The associate editor coordinating the review of this manuscript and approving it for publication was Moayad Aloqaily.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

by binding laws and regulations, trade based manipulation in stock prices are harder to eradicate [4]. It implements a variety of strategies like quote stuffing, market close, wash trade [4], pump and dump [5], ramping or gouging also known as Spoof Trading [6] etc. Several previous researches mentioned in the review section made few attempts in this field using both labelled and unlabeled datasets [3], [7], [8] but failed to acknowledge the rare and expensive labelled dataset, diverse detection model for multiple manipulation schemes and the heuristically assumed values for the model parameters involved in the decision making process. The major contributions of this work are as follows:

The idea for anomaly detection is to generate an adept model of the data distribution that can establish a clear manifold between normal and abnormal data instances. This research proposes the combination of distribution modelling approach using kernel techniques and non-linear transformations technique onto higher dimensions in order to create linear manifolds among data points. For non-linear data analysis, KPCA is used to project the original dataset onto higher dimensions, sorted as per their variances. Once the KPCA forms a non-linear boundary among the transformed data in higher dimensions, the first step of the detection model is implemented. Although, one of the conventional approaches for calculating such transformed feature vectors aim to compute the reconstruction error and forms isopotential curves as the decision boundaries which is limited by the highly computational complexity [9], [10]. A rather simpler approach is to limit the number of extracted feature vectors (principal components) and to subject them onto the proposed multidimensional kernel density estimation (MKDE) based clustering algorithm for further evaluation in the second step. The proposed MKDE clustering helps in grouping the data into clusters (only normal trades) without asking the number of clusters up front [11]. The major advantages of using this approach is its decision-making capability based on analyzing the patterns that are subjected being an anomaly without prior information about the location or the nature of the manipulation and also helps in reducing the total amount of computations. This can be achieved by clustering the data, without asking for the number of clusters upfront using the proposed clustering algorithm, which is now linearly separable due to KPCA transformation and marking the data points left unclustered as anomalies. A dataset involving thirteen different stocks intraday price information from multiple resources (both UK and US stock exchanges) and three distinct manipulation schemes are considered for an exhaustive evaluation of the proposed approach. A distinctive comparison of the proposed approach with the existing benchmark approaches and conventional anomaly detection techniques indicates a significant improvement in terms of detection accuracy, F-measure and a substantial fall in the false alarm rates. In order to check the validity of the approach in terms of non-stationarity, stock price data from both UK and US leading stock exchanges are considered.

The rest of the paper is organised as follows: Section 2 describes different price manipulation schemes using real life examples. Then, existing relevant manipulation detection techniques are reviewed in section 3. Subsequently the proposed approach is presented in detail in section 4 followed by an experimental evaluation in section 5. Section 6 presents an analysis and discussion of the obtained results. Finally, section 7 concludes the paper and identifies future research directions.

II. STOCK PRICE MANIPULATION SCHEMES

Stock price manipulation refers to artificially influence market prices of stocks using illegitimate means. The intention is to manually, though illegally effect the current market price of a stock for potential benefits. Market manipulators tend to influence stock prices using a variety of manipulation schemes. Few of such schemes covered in this approach are traditional pump and dump and emerging high-tech schemes like spoof trading/ramping, quote stuffing. They are selected because of their impact on the market and the increasing number of cases SEC put to trial [12].

One of the most prominent type of price manipulation tactic is Spoof Trading [6] also known as ramping. As an example, a manipulator wants to sell a stock at a higher price than the current ask price. The manipulator will enter spoofed buy order in a larger volume at a higher price than the current bid making other investors believe that this increased price is genuine thus expecting other legitimate investors to join. Once the order is matched, the manipulator will withdraw the large spoofing buy order then issue a sell order of large volume of shares at this manipulated price as shown in Fig 1(a). A manipulative spoofing order stays in the grey zone until disclosed, as the orders mentioned in the order book cannot guarantee which of them is real or fake. In the case of pump and dump, the manipulator begins by creating a high demand of a stock using false information [5] leading to its price rises (pumped) and the manipulator sells it (dumped) when sufficient number of orders are added or when the desired bid price is achieved shown in Fig 1(b).

A similar trading manipulation scheme is quote stuffing, which usually happens in high frequency trading (HFT), where the manipulator uses high frequency trading algorithms to flood the market by quickly entering and withdrawing a large number of non bona-fide buy and sell orders [13]. This hereby creates a confusion among the traders about the amount of trading activity. This further affects the normal investors in delaying their trades especially the participants that do not use HFT algorithms and consumes a lot of exchange resources [14]. One of such a case study has been presented in Figure 1(c). It can easily be comprehended that the number of trades fell to a lowest level during the time interval (651 seconds ∼11 mins) when abnormally large amount of quotes/sec (∼10000 plus) were made [15]. As illustrated in Figures 1(a-c), most price manipulation activities follow a trend of increasing the price of a stock by submitting non-bona fide orders, executing the sell at the

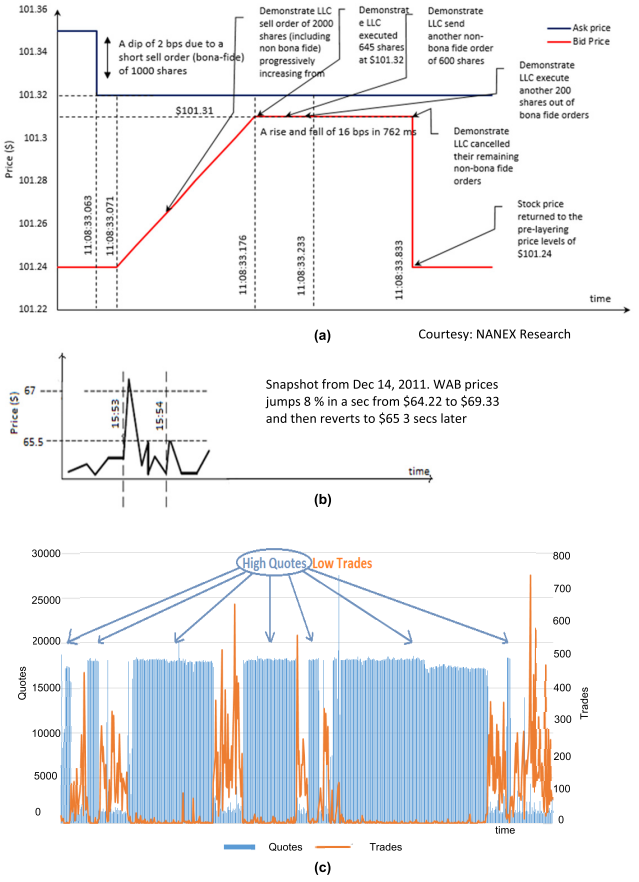

FIGURE 1. **(a) Illustration of the spoofing activity (Saw-tooth waveform) on 25th Sept 2012 and (b) Snapshot of the pump and dump (Spike waveform) manipulation activity from Dec 14, 2011 shows an 8 % rise of Westinghouse Air Brake (WAB) Tech. price within 1 sec and return to previous level 3 secs later. (c) Snapshot of Quote stuffing activity from 01st Nov 2012 shows thousands of quotes been sent to flood the market from 12:26:50 to 12:39:42 pm. As is observed the number of trades fell to a lowest level during this interval.**

manipulated price and then a rapid withdrawal of the buy order leads to a sudden drop in prices as well. As stated before, the implication of manipulation schemes like spoofing trading, ramping and pump and dump can be critical on the market [16]. A detailed representation on Spoofing shown in Figure 1 frames up the rise and fall of prices for Demonstrate holdings LLC listed on NYSE in a total span of 1.3 secs [17]. The sale was executed at $101.32, which is around 8 bps up than the current bid price as shown in Figure 1(a). Another manipulation case of pump and dump is illustrated by a spike pattern on Westinghouse Air Brake (WAB) Technologies Corp. where the manipulated bid price is moved 8% and reverted to its prior level in tiny time interval of 3 secs as shown in Figure 1(b) [18].

A detailed survey report presented in [19] provides an insight into the modelling techniques used in financial data. Along with prediction, a vast number of research studies have been carried out on stock market manipulation detection. Since the financial crisis of 2008, Volatility Index reaching record levels, the flash crash of 2010 [20], [21] and because of the abusive activities, markets have been highly monitored by market analysts, regulatory organisations and researchers. The following section will review existing research studies that have been conducted in trade-based manipulation that used both supervised and unsupervised Machine Learning (ML) techniques.

III. RELATED WORK

A vast number of empirical studies have been conducted in stock price manipulation but most of them claimed significant improvements in the detection results either based on certain assumptions in their applied research or using labelled datasets, which makes it easier for the model to learn the anomalous patterns and provide better detection accuracy on the test data.

Yang *et al.* [22] constructed a prediction model for the detection of stock price manipulation activities using PCA followed by a logistic regression representing the discrimination for the manipulated stock prices. However, the results obtained do not highlight the detection performance rather the prediction of the stock prices is given only. A market close ramping detection algorithm developed by Aitkens *et al.* [3] presented an empirical study of relationships between the market efficiency and the manipulations detected by the algorithm. The algorithm was able to detect manipulations according to the historical price change when it exceeds a threshold that was set as the 99% histogram distribution cut off of the historical price change during the corresponding time slot near the close of the trading session. A case study based on manipulated stocks of Dow Jones Industrial companies from 2003 was considered to identify suspicious trading activity in relation to stock price manipulation by Golmohammadi *et al.* [23]. Experimental results show that the proposed approach outperform other learning methods such as kNN, C5.0, neural network achieving an F2 score of 53% and out-performing them by a huge margin of 30% in sensitivity but fails to reduce the false positives.

Ögüt *et al.* [24] compared the performance of Probabilistic Neural Networks (PNN) and Support Vector Machines (SVM) with statistical multivariate methods like Discriminant Analysis and Logistic Regression. The dataset from Istanbul stock exchange (ISE) used in this research was labelled for normal and manipulative content making it suitable to employ supervised learning techniques. Results proved that popularly used machine learning techniques like artificial neural network (ANN) and SVM performed better as compared to the statistical multivariate analysis in terms of classification accuracy. In order to further improve the performance of a neural network, Leangarun *et al.* [25] implemented a two-step method for the calculation of the feature set and then used a feed-forward neural network model for detecting pump and dump and spoofing manipulations. The dataset from the LOBSTER project [26] used by the model is a combination of level 1 and 2 at the depth of the order book consisting of labelled data, normal trades from level 1 and manipulative ones from level 2. The model achieved 88.28%

accuracy in the detection of pump and dump case but failed to identify the spoof trading case effectively.

Cao *et al.* [27] proposed a novel approach for stock price manipulation including ramping and pump and dump using Adaptive Hidden Markov Model with hidden states as anomalies (AHMMAS). The method claims an improved performance in terms of the area under the ROC curve and the F-measure, for the four features proposed over other classification techniques like One Class SVM (OCSVM) and kNN. Although, this research aimed to provide better detection capability for an anomaly in the financial data, it relied on the assumption that data is generated from a particular distribution and used semi-supervised training for HMM by calling normal and abnormal instances from the GMM distribution. However, this assumption often does not hold true, especially for high dimensional real data sets but could have been justified by using several hypothesis tests which could have been added in the research. Again, the derivative feature set used in this research are not calculated as per the definition rather just as the differential of the variable with time but did not considered the time gap between any two consecutive samples. The approach focused on decomposing the data using Dirichlet Process Gaussian Mixture Model (DPGMM) into different components defining normal and abnormal components and then trained a Markov model upon those components. Furthermore, the research specified the number of decomposition components, which is misleading as the distribution of the normal-abnormal patterns, might overlap with each other, if the specified number is less or more.

Diaz *et al.* [28] analysed and compared the knowledge discovery techniques of data mining such as linear and logistic regression for stock price manipulation. They modelled the returns, liquidity, and volatility as well as the news and events related to the stocks using logistic regression function defined. Although, the authors claim to detect stock price manipulation (inclusive to any specific scheme) using unsupervised learning over market moves like trading volume effects, liquidity and returns as part of a quantitative analysis, no account of specific unsupervised techniques used were mentioned. The authors, however, used intra-day stock data but considered average returns, average volume and average volatility rather than tick features that again make it difficult to specifically locate anomalous data. This knowledge gap between the statistical features and detection techniques leads to irregularities in the manipulation models developed and hence is prone to suffer from a higher error rate even for the legitimate trading activity. The authors also trained several supervised classifiers like C5, QUEST and CR&T for the same feature set and achieved higher detection results (Accuracy = 93%) but used no proper labelling in terms of the timing instances for manipulative data, as the time frame for manipulation from SEC proceedings was highly vague. Also, a subsequent analysis of the manipulation results was also missing from the work.

Ferdousi and Maeda [29] applied an unsupervised learning approach called peer group analysis (PGA) to the stock manipulation and detected cases of manipulation with an acceptable rate of detection. However, they did not consider the change of peer groups over time, which decreases the detection probability when some members in the same peer group may gradually exhibit distinct behaviour from that of other members. Kim and Sohn [30] extended this concept and tried to improve PGA approach by updating the size of the group with time and achieve acceptable detection accuracy (AUC = 0.845) but failed to identify the exact location in time of the suspicious activity. Although they tried to generalise the concept of anomaly in financial data rather detecting individual schemes, a subsequent step should have been added to identify the type of the manipulation activity. Most of the manipulation schemes follow a sequence of patterns rather than a single event that can be identified as an anomalous behaviour, an aspect that is also missing from this approach. Recently, Wang *et al.* [31] proposed the use of recurrent neural network (RNN) for stock price manipulation detection. The research proposes to leverage the RNN ensemble learning model by using trade based features combined with characteristic features of the stocks implemented. The dataset used in this approach are taken from Shanghai stock exchange, China. The research attempts to detect manipulation instances by training an RNN model using ensemble learning while following feature selection, modelling and prediction using labelled dataset. It also claims to outperform traditional methods by 29.5% in terms of AUC. Although, the proposed model made use of labelled dataset but failed to specify the manipulation schemes detected. Use of supervised learning approach makes the detection model biased and is always vulnerable to failure whilst a contemporary manipulation scheme is present.

It is evident to state that many of the past researches using data sets having manipulated samples prosecuted by SEC or synthetically created, false detection rates for many of the proposed approaches have not been evaluated, which also challenges the integrity of the features used. Moreover, the success of the existing models was based on specific data sets and the lack of adaptive capability to capture the different manipulative strategies. To summarise, following issues in the previous researches are the major challenges in designing a detection algorithm for trade-based manipulation.

- Use of labelled datasets which is difficult to acquire as it is rare and expensive. It further makes the detection model biased towards the given dataset.
- Focus on specific manipulation scheme and the choice of specific parameter values necessary for the detection of the chosen manipulation scheme. This makes the proposed model biased towards a particular manipulation pattern rather diverse and lacks the adaptability towards other manipulation schemes.
- Most of the approaches have focussed on a limited number of stocks listed on a local exchange rather than platforms like NASDAQ and LSE where stock prices are effected on a global scale.

In this research, problem formulation is carried out by analysing original stock price information, then adding three types of synthetic anomalies that correlate with the existing manipulation activities into it. Although the original data is injected with three types of anomalies per stock, a significant number of them are added at different time instants along the duration of the time series to check the robustness of its detection capability. The following section explains the methodology of the proposed approach followed by experimentation with the dataset having added anomalies and compares its performance with the some of the existing approaches mentioned above in the later sections.

IV. METHODOLOGY

The concept of manipulation detection in financial data revolves around the fact that since in a time series, several attributes of anomalous trading transactions overlaps with normal ones [32], proper characterization of manipulation used is required. It makes sense to state that since the stock price data is non-stationary in nature, elementary properties including mean, variance and correlation varies over time. Such variations can be related with the economics of the market microstructures [33]. Hence, the intention of the proposed model is to derive a set of features linearly independent or uncorrelated from each other when transformed in orthogonal dimensions. It should be kept in mind that since financial data is not sparse in nature [34], a large computational complexity is involved with the conventional approach of orthogonal transformation by calculating the iso-potential curves or surfaces of the reconstruction error. To avoid this, input data is divided into a series of a particular length windows, followed by the proper selection and adaptation of the transformed orthogonal features. A second step of clustering based technique is then applied on to such orthogonal dimensions to identify the abnormal samples. Hence, the methodology of this research follows a two-step approach: Firstly, the input feature set is extracted based on the concept of capturing manipulated patterns and projected onto higher dimensions. Secondly, focus is laid upon to carefully select and adapt the features from the transformed domain. Finally, anomalous stock prices/trades will be detected by using multi-dimensional clustering techniques to cluster normal and abnormal trades.

A. FEATURE CHARACTERIZATION

As for any dataset, the amount of redundancy can be reduced only if relevant information is extracted from it. The dataset used in this research are the stock prices of thirteen different companies operating at NASDAQ and London stock exchange (LSE) from multiple sources. As high frequency components in financial data are more prone to manipulation activities, focus is laid upon extracting relevant features that can capture the effect of high frequencies along with other attributes like derivatives [35] and differences. For time series (stock prices) that consists of synthetically added manipulated samples, denoising techniques [27] is applied using wavelet transform. This is done to filter out the low frequency components in the data and the filtered output is used as a feature, $\hat{x}(t)$ where $x(t)$ is the input time series (stock prices). This is calculated by applying discrete wavelet transform (DWT) on the input data decomposing it up to first level [36] into detail and approximate coefficients. Detail coefficients represent high frequency components and approximate coefficients represent low frequency components. Furthermore, in detail coefficients $X_{a,b}$ where a & b are scaling and shifting parameters, to extract only top high frequency components, hard thresholding is inversely applied for a selected threshold γ such that the components exceeding the threshold are set to zero. The value of γ is selected using universal threshold algorithm [37].

$$X_{a,b} = \begin{cases} X_{a,b} & X_{a,b} \leq \gamma \\ 0 & X_{a,b} > \gamma \end{cases} \tag{1}$$

Inverse DWT is then applied on the detail coefficients so obtained and approximate coefficients to reconstruct the time series $\widehat{x}(t)$, here. Along with this, a set of five individual features are also used where most of them computes the change between the two consecutive data instances that can help stabilize the mean and also helps in minimizing the trend and seasonality of the data. List below describes the feature set used:

1. Input price series, $f_1 = x\,(t)$.
2. High frequency component, $f_2 = \hat{x}\,(t)$.
3. Wilson's amplitude [38], $f_3 = w\,(t)$,

$$s\,(t) = x\,(t) - x\,(t-1)$$
$$w\,(t) = \begin{cases} 3 * s\,(t)\,, & s\,(t) > threshold \\ s\,(t)\,, & s(t) \leq threshold \end{cases}$$

Here, $s\,(t)$ is the difference between two consecutive samples. Typically, a threshold value of 3 bps is selected.

4. Derivative of the input stock price [35], $f_4 = \frac{\partial x(t)}{\partial t}$.
5. Gradient of the feature set containing high frequency components, $f_5 = \frac{\partial \hat{x}(t)}{\partial t}$.

B. KERNEL PRINCIPAL COMPONENT ANALYSIS (KPCA)

Principal component analysis (PCA) is the orthogonal projection of data into lower dimension linear space such that the variance of the data in each projected dimension is maximized [39]. Despite PCA's ability to project the data onto lower dimensions, its interpretability remains confined by the fact that components generated by standard PCA have added noise and exhibit no meaningful pattern that can be either well represented or visually observed in a linear subspace [40]–[42]. We propose the use of kernel PCA in financial data as it is essential here to uncover localised stock price microstructure patterns using non-linear transformations in higher dimensions that could account for main variability in the temporal data. The role of KPCA also becomes crucial in avoiding the vulnerability of the stock prices during the long held position of stocks that sometimes introduces

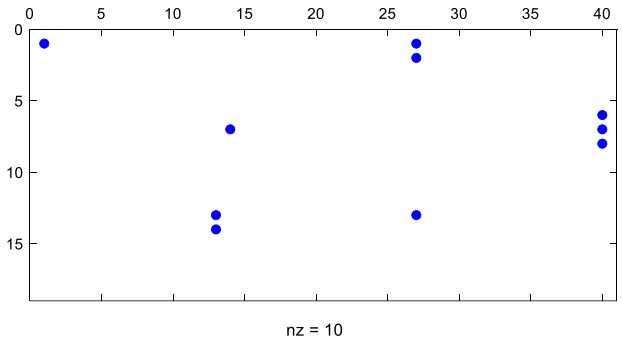

FIGURE 2. Sparse adjacency matrix representation for Apple Stock feature set $F_0 \in [720 * 5]$ from 12:20:05 PM to 12: 21:19 PM on 21st June 2012. 'nz' represents the number of non-zero elements present for that duration. Total number of data instances included = 720; axis as the rows and columns of the adjacency matrix generated for the weighted graph of the input F_0. The 'blue' dots shown in the figure represents the connections that can only be established for non-zero elements within the graph matrix.

sparsity in the feature set. Figure 2 shows the sparse adjacency matrix representation for such a situation with Apple stock from 12:20:05 PM to 12:21:19 PM on 21st June 2012. It shows the connections among non-zero elements within a graph adjacency matrix [34] that can be generated for the considered feature set $F_0 \in [720 * 5]$.

Kernel PCA uses a non-linear transformation of the input data having d dimensions to the m dimensional space ($d \ll m$) using kernel methods [43]. It does so by mapping the input data points to a higher dimension feature space using kernel trick, forming a linear/non-linear hyperplane and then reconstructing the data set in the decreasing order of their variances using standard PCA.

An input feature vector, $\boldsymbol{x_i} \in \mathbb{R}^{\boldsymbol{d}}$ ($d = 5$ and $i = 1, 2, \ldots, N$) having N number of input data instances in the feature set $F_o^5 = \{f_o^1, f_o^2, f_o^3, f_o^4, f_o^5\}$ is first transformed to a higher dimension feature space F_t^m (for m dimensions in the mapped space) using a non-linear transformation, $x \rightarrow \varphi(x)$ where φ is a non-linear function. The kernel trick, herein suggests the calculation of extracted features (principal components), covariance matrix (Eq. 2), and subsequently Eigenvectors and Eigenvalues in the transformed domain F_t^m, is possible without calculating the intractable transformation or mapped data point $\varphi(x_i)$ of a given input data instant, x_i [44].

$$\begin{aligned} C^{F_t^m} &= E_x[(\varphi(x) - E_x[\varphi(x)])(\varphi(x) - E_x[\varphi(x)])'] \\ \text{Or} \quad C^{F_t^m} &= E_x\left[\tilde{\varphi}(x)(\tilde{\varphi}(x))'\right] \end{aligned} \tag{2}$$

where, $\tilde{\varphi}(x)$ is centred at the origin or a zero mean vector of the transformed/mapped data points. In the feature space F_t^m, Eigenvector V of the covariance matrix, $C^{F_t^m}$ can be defined and there are coefficients α_i such that,

$$V = \sum_{i=1}^{N} \alpha_i \tilde{\varphi}(x_i) \tag{3}$$

Recalling the Eigenvalue and Eigenvector relationship from a standard PCA, we can write,

$$\lambda V = C^{F_t^m}.V \tag{4}$$

Note that $C^{F_t^m}.V$ is a dot product and λV is a scalar product where λ being the Eigen value of $C^{F_t^m}$. The length of α can be calculated from the normalisation of Eigenvectors, $V.V^T = 1$ or $\|V\|^2 = 1$. Using (2), (3) and (4), $\|\alpha_i\|^2 = {}^1/_{\lambda_i}$. Now, the Eigenvector V can be calculated by defining a Kernel matrix as the dot product of two feature points in the mapped space F_t^m,

$$K_{i,j} = \tilde{\varphi}(x_i).\tilde{\varphi}\left(x_j\right)' = \tilde{k}(x_i, x_j) \tag{5}$$

$\tilde{k}$, can be further defined as the kernel function to calculate the inner product and can be substituted with the most commonly used radial basis function (RBF),

$$\tilde{k}\left(x_i, x_j\right) = exp\left(\frac{-\left\|x_i - x_j\right\|^2}{2\varepsilon^2}\right) \tag{6}$$

For $\tilde{\varphi}(x_i), \tilde{\varphi}\left(x_j\right) \in F_t^m$ and ε being the kernel bandwidth parameter, kernel components are amplified given their density estimate falls below 10% of its maximum value (7). This is done in order to increase the spread between the normal and abnormal trading prices in the kernel space, the effect of which can be seen in the transformed feature space.

$$\tilde{k}\left(x_i, x_j\right) = \begin{cases} 3 * K_{i,j} & \mathcal{P}^{K_{i,j}} < 0.1 * \max(\mathcal{P}^{K_{i,j}}) \\ K_{i,j} & \text{otherwise} \end{cases} \tag{7}$$

where $\mathcal{P}^{K_{i,j}}$ is the density estimate of the data points in kernel space. By substituting (6) and (2) in (3), the Eigenvectors and values can be calculated. The projection of new data points onto the mapped Eigenvectors or the principal components in F_t^m is given by, $t_i = \varphi(x_i).V$ which can be further simplified by solving (3), (4) and (5). The objective is to visualize the data points in the kernel space, increase the spread among data points and forward this effect onto the transformed space.

Some of the major constraints in the implementation of KPCA are the choice of RBF kernel parameter and the number of principal components to be considered. It is well documented that for anomaly detection using PCA, the number of components extracted l should be such that the cumulative variance must be greater than or equal to 90% of the total variance in the mapped feature set F_t^m [22], [45] which settles down to $l = 7$. This helps in reducing the uncertainty over the optimal size of the components used and will efficiently reduce the computational complexity of the overall approach. To deal with another constraint about the selection of the kernel parameter, an efficient method is to keep the value of ε fixed for a given input data [46]. The choice of ε is carried out in such a way as it maximizes the amount of variance for the considered number of principal components and minimising the reconstruction error for the projected feature space as proposed in [47]. Figure 3 shows the components extracted from KPCA applied to a set of

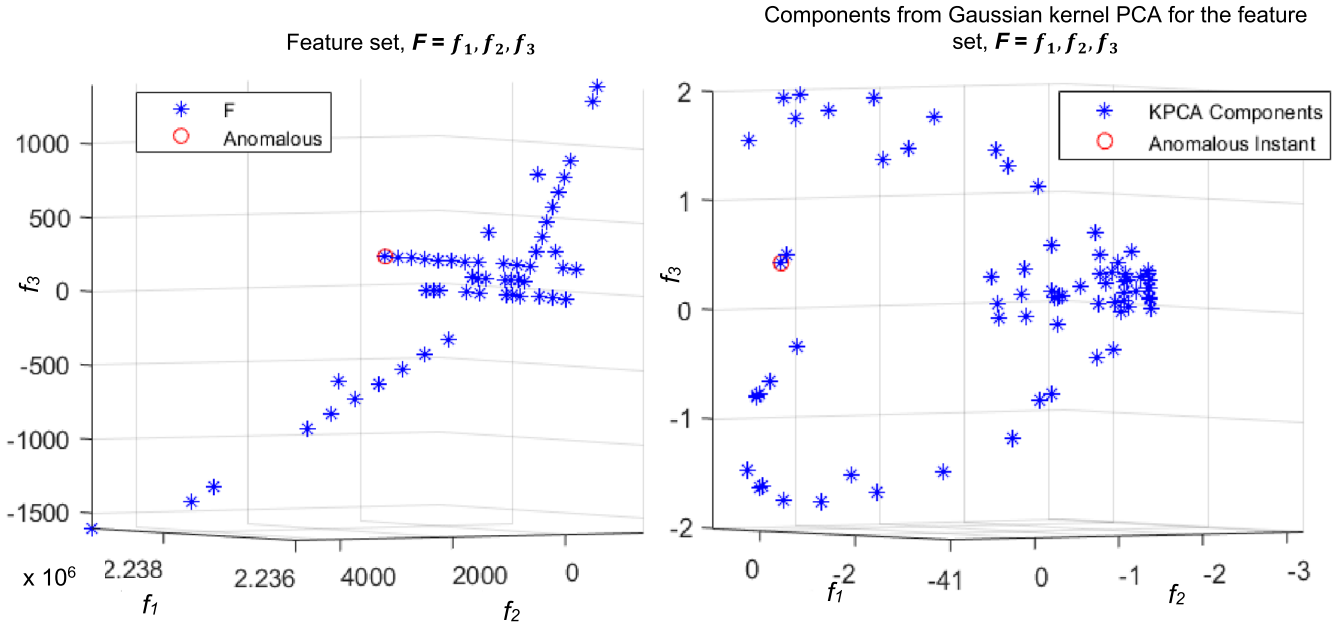

FIGURE 3. Components extracted from input feature space in $\mathbf{R^3}$ using KPCA including normal and anomalous data instances.

five features for Apple data after normalization (for clear observation only first three components have been shown out of $d = 5$ in this case). The dataset used, enclosed both normal and anomalous stock prices.

C. MULTI-DIMENSIONAL KERNEL DENSITY ESTIMATION

Multi-dimensional Kernel Density Estimation (MKDE) clustering based anomaly detection is a modified approach for anomaly detection via non-parametric density estimation for clustering [11]. It has the advantage that it does not require a priori knowledge of the number of clusters. The method suggests calculating a kernel based probability density estimation for a set of data samples and cluster them based on the following algorithm [11]. For an input data sample 'F_t^m',

$$F_t^m = \{f_1, f_2, \ldots f_m\}^T$$

The kernel density estimator used to calculate the probability density $\hat{P}(f)$ is given by

$$\hat{P}(f; h) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{f - \bar{f}_i}{h}\right) \tag{8}$$

where 'm' is the number of dimensions of the data to be clustered, $\bar{f}_i$ is the mean of i$^{\text{th}}$ data sample for a total of n instances, $f_i = \{f_{i1}, f_{i2}, f_{i3} \ldots f_{im}\}^T$ and 'h' is the smoothing parameter or bandwidth for m-dimensional input data. The selection of such a smoothing parameter forms an important entity in MKDE estimation. It is seen that for the same dataset, different bandwidth can have serious effects on the results [48]. The kernel function $K(x)$ is calculated via a linear diffusion process [48] leveraging a Gaussian kernel density estimator (Eq. 9) as it lacks the local adaptive behaviour towards outliers [49], resulting in misleading bumps and hence flatten peaks and boundary bias. Although such problems can be solved by using high order Gaussian kernels [50] but they are unable to provide proper non-negative density estimates [51].

$$K(x) = \left(\frac{1}{2\pi}\right)^{-d/2} exp\left(-\frac{x^T x}{2}\right) \tag{9}$$

Given: $x = \{x_1, x_2, \ldots x_q\}^T$ be a $q * m$ size dataset that is to be clustered after using KPCA upon five input original features and $x_i \in \mathbb{R}^m$; the parameterisation of the bandwidth matrix h as a diagonal matrix [52] is optimised again via diffusion estimator in [48] and evaluated using Asymptotically Mean Integrated Square Error (AMISE) [53].

D. DETECTION ALGORITHM BASED ON MKDE CLUSTERING

The algorithm for MKDE clustering works by first calculating the kernel density estimate for a given dataset using an adaptive smoothing parameter (h), defined in the previous section also known as bandwidth. For a given set of data instances, if the difference between the mean of the estimate and the data values is less than the bandwidth, the given sample points are grouped into a cluster. For the remaining data points having a new estimate, the difference is again calculated, samples having difference less than the bandwidth (of the dataset under consideration) are again grouped into another cluster and the process continues. The algorithm is originally designed to deal with univariate data [11]. As there are seven dimensions extracted from KPCA in the financial feature F_t^7 set used here, seven separate smoothing parameters are obtained for each dimension. The first problem that should be tackled with is the estimated multi-modal PDFs in

multi-dimensions. In such cases, when multi-modal PDFs are generated, it is difficult to determine one mean value for the whole set of data points, as there are several means generated for the given distribution, as shown in Figure 4 (each peak in a multi-modal distribution). Now, each mean is considered separately along with multiple smoothing parameters for each dimension. The implemented algorithm can lead to multiple clusters even within a compact group of data points. In addition, the cluster values for one may overlap with the one adjacent to it (depending upon the value of the bandwidth selected). In such a situation, the clusters that overlap must form a single cluster, but not if one of them is an anomaly as shown in Figure 5. Such problems are quite common while dealing with anomaly detection in financial data [55]. It is therefore necessary to define a highly illustrious feature that can resolve the two clusters separately and adapt the clustering approach in this case. It should also be noted that there are a number of methods (cubic spline, gradient ascent etc.) to cluster this kind of dataset but very few to distinguish between normal and abnormal data, which makes such a problem of clustering based anomaly detection, a challenging task. Algorithm 1 presents the possible solution to resolve such an issue of the multi-modal distributions and the insights of the price manipulation detection process.

Algorithm 1 Stock Price Feature Clustering and Anomaly Detection

1. For any specific stock, extract the feature set $F_o^5 = \{f_o^1, f_o^2, f_o^3, f_o^4, f_o^5\}$
2. Apply KPCA on the features considered and transform them into, $F_t^7 = \{f_t^1, f_t^2, \ldots f_t^7\}$
3. For a selected window of samples (F), construct their joint probability distribution $\hat{P}(F, t; h)$; $F, h \in \mathbb{R}^7$ using multi-dimensional KDE approach [54] for bandwidth (h).
4. Construct the MKDE based clustering model for anomaly detection:
   a. Given: $F_t^7 = \{f_t^1, f_t^2, \ldots f_t^7\}$, for $f^i \in \mathbb{R}$ is the feature sample.
   b. Set: $\mathcal{C} = \emptyset$, $t =$ length (f); where $\mathcal{C}$ is a cluster.
   c. j = 0;
   d. WHILE $F \neq \emptyset$; % F is the set of data samples to cluster
      j = j + 1; % Iteration counter
      % define the bandwidth h and $\bar{f}$ is the mean(s) location of distribution for the data samples
   e. FOR i = 1,2,3.....t
      IF $|\bar{f} - f| < h$
      $\mathcal{C}_j = \mathcal{C}_j \cup f_i$; % Add the set of data for all the features f_i to the Cluster $\mathcal{C}_j$
      $f = f \setminus f_i$; % Remove the clustered data from the original set
      ENDIF
      ENDFOR
   ENDWHILE
5. In case of Multi-modal PDF as shown in Figure 3, for the so-called clusters formed,
   FOR $\mathcal{C}_i = \mathcal{C}_1 : \mathcal{C}_j$ % for j, number of clusters
   $d' = \min \|\mathcal{C}_i, \mathcal{C} \setminus \mathcal{C}_i\|$;
   IF $d' < h \&\& \mathcal{C}_i \cap \mathcal{C} \setminus \mathcal{C}_i = \emptyset \&\& \frac{\mu(\hat{P}_{\mathcal{C}_i})}{\mu(\hat{P}_{\mathcal{C} \setminus \mathcal{C}_i})} < 0.7$
   % For every cluster $\mathcal{C}_i$, if it is not overlapping with the rest of the clusters $\mathcal{C} \setminus \mathcal{C}_i$, and if the ratio of their individual PDF at their respective means is less than 0.7 i.e. if the ratio is greater than 70%, they will be treated as separate clusters or else combined into one.
   $\mathcal{C}_i = \mathcal{C}_i \cup \mathcal{C} \setminus \mathcal{C}_i$; % Merge the clusters
   $\mathcal{C} \setminus \mathcal{C}_i = \emptyset$;
   ENDIF
   ENDFOR

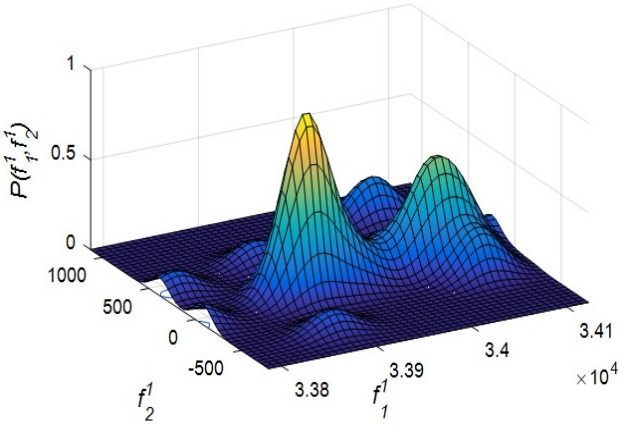

FIGURE 4. Bi-Modal PDF i.e. having two means shown for a subset of features $\{f_1^1, f_2^1\}$ of Apple stock for considering 100 samples from 9:30:01 AM to 9:30:01.05 AM.

After formal implementation of the above algorithm, two critical situations may arise in this case. First, if the number of left out data points considered are fairly large and more than one anomalous value in the distribution so obtained (forms a cluster of their own, given their separation, d' is more than the bandwidth). Such a problem can be avoided by using robust features and selecting a proper window size under consideration. Second, if the data instances are sparse as shown in Figure 2, it is a possibility here that an anomalous trade may be clustered with the normal ones. To address such a situation, KPCA helps in reducing the sparsity of the dataset and is adapted to increase the spread among normal and abnormal data instances. The data instances that are not clustered are marked as anomalies. It should also be noted that the above described process is not totally focussed on devising a new clustering method, but rather an approach to narrow down anomaly detection problem.

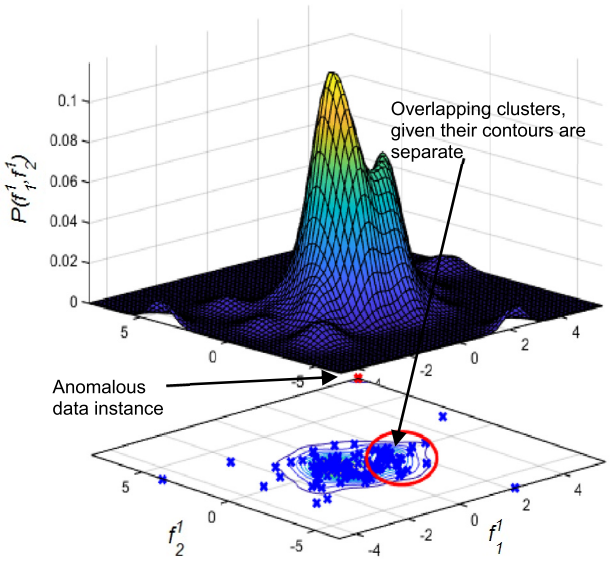

FIGURE 5. Probability distribution using kernel density estimate for a 2-D feature set ϵ $\{f_1^1, f_2^1\}$ of Apple Stock for 100 data points along with its contour.

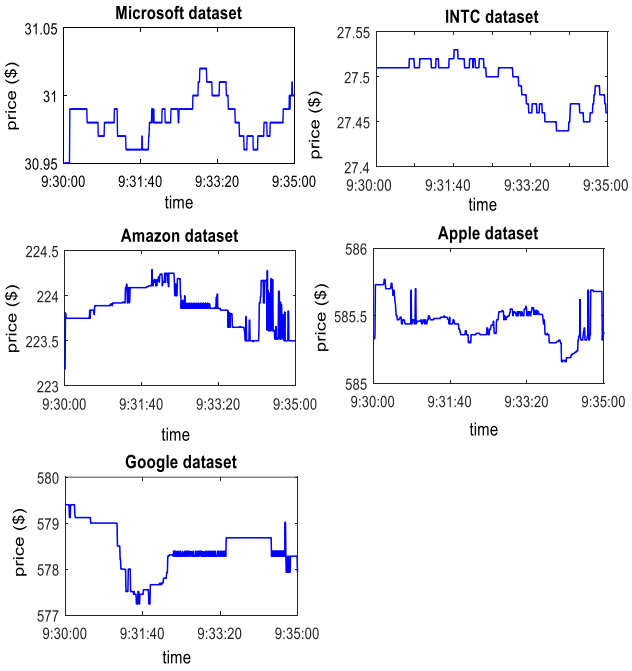

FIGURE 6. Varying bid prices of different stocks from 09:30:00 AM to 09:30:52 AM on 21st June 2012.

V. EXPERIMENTAL EVALUATION

A. DATASET USED

The dataset used in this research comprises of thirteen different stocks including Apple, Amazon, Google, Intel Corp and Microsoft for 21st June 2012 and others including Apple, Amazon, Microsoft, Google, Intel Corp, EBAY, Cisco, Netflix, Nvidia, Facebook, SIRI US, QUALCOM and AMD from 12th November 2018. It consists of level 1 tick data of stock price information along with its derivative for 21st June 2012 on NASDAQ Stock Exchange, USA taken from the LOBSTER project, [26] and the stocks from 12th November 2018 taken from Bloomberg trading platform, Newcastle business school (NBS), Northumbria University, Newcastle, UK. Figure 6 shows the variation of the bid price for different stocks beginning from 9:30:00 to 9:30:52 from 21st June 2012. Such a data is selected for its high volatility, high trading frequency and the total number of trades per day (∼1 million per stock) that makes it prone to manipulation as aforementioned. The database from the LOBSTER project, employed in this research is free from any manipulation activity [9], [27]. Hence a synthetic dataset is prepared by injecting artificially generated anomalies similar to the ones shown in Figure 1 into the data stream making it a combination of normal and manipulative trades. Since, the dataset collected from NBS has not been reported to have price manipulation yet; the results calculated from such are not compared with the existing researches in stock price manipulation detection.

B. EXPERIMENTAL SETUP

The dataset varies in the size of each stock used, based on how they have been categorized into two groups; *Group I* having Apple, Amazon, Google, Intel Corp and Microsoft stocks, each converging itself within the range of 200,000 samples to a bit more than 800,000, for any one form of trade (Ask or Bid) from LOBSTER project. *Group II* having the stocks taken from NBS Bloomberg trading platform having more than 1 million trades in Bid/Ask for a given day. Prior to using *group I*, it is made sure no abnormal trading activity was detected [27] and reported by any regulatory organisation for these stocks on the given day [9], marking it as a normal dataset without any manipulation. In order to check the robustness of the proposed approach, three different types of anomalies as shown in Figure 1 complementary to the real life scenarios are injected [17] into this time series in significant amounts. Number of anomalies injected are also varied based on the size of stocks in each *group*. As the size of *group I* stocks varies considerably, it has been sub-categorised into *Group A* for Apple, Amazon and Google stocks as the average number of trades are limited to 200,000 and *Group B* for Intel Corp and Microsoft stocks having the average number of trades approximately equal to 800,000. Following this premise, *group A & B* stocks are injected with 100 and 200 anomalies/type, respectively making a total of 300 and 600 anomalies per stock with considerable spacing among them. For *group II*, since the size is almost comparable with Group B stocks, 200 anomalies of each type are injected in every stock making it 600 anomalies per stock. Such a configuration of synthetic data is practically accepted as per the business standards [56] and is then tested for the proposed model. To ensure comprehensive assessment of the approach, the detection is performed without a priori information about the location, amplitude and time span of the anomaly injected. It is also possible that a given anomaly will be followed by

a rather similar, non-anomalous, waveform in shape but any prior knowledge about any succeeding or preceding samples is totally avoided. Once the transformed feature vectors $\boldsymbol{F}_t^7$ are obtained from KPCA, they are windowed into a heuristic sample size of 500: $\boldsymbol{F}_t^7 = \{f_t^1, f_t^2 \ldots . f_t^7\}$ for $t \in [t, t + 500)$ and are then supplied to MKDE clustering algorithm for manipulation detection. Such a condition is further explored, and the detection results are calculated by varying window sizes. Furthermore, to improve the robustness of the approach, the displacement between the added anomalies is varied to check how the model reacts, if two different anomalies are placed closed to each other.

Most of the proposed approaches described in Section 3 have claimed a considerable amount of detection accuracy in price manipulation. As some of the models [3], [23], [28] focussed on the detection of a specific manipulation scheme rather than presenting a general detection model, an adept comparison with such proposed approaches is avoided. However, advance computational models like AHMMAS [27], Naïve Bayes based model [23], Probabilistic Neural Network (PNN) [25] and Peer group analysis [29] were selected as the benchmark approaches for the proposed model. An evaluation metric defined for the representation of the results is Receiver Operating Characteristics (ROC) curve and the Area Under its Curve (AUC) [27], [57], [62], [63]. It is also worth mentioning that although ROC curve evaluation is often used with classification approaches trained using labelled data, there are various instances of it being used in totally unsupervised approaches [57]–[63].

For the experimental setups described in this section, the following section discusses the obtained results and analyses the pertinence of the model in manipulation detection.

TABLE 1. AUC comparison of proposed approach with benchmark approaches.

	KPCA-MKDE	NB [20]	PNN [22]	AHMMAS [27]	PGA [29]
Microsoft	**0.9143**	0.8560	0.7977	0.7336	0.8289

VI. RESULTS AND DISCUSSION

The ROC curves for five different stocks having added anomalies (300 and 600 for *Group A & B* respectively) are shown in Figure 7. The tweaking factor that varies TPR and FPR values is the threshold applied on the output score. Here, the output score of the proposed approach is the difference between mean of each cluster and the corresponding sample that ultimately leads to the decision as to whether a sample is manipulative or normal. AUC values in table 1 for some of the existing benchmark approaches in stock price manipulation detection are calculated only for Microsoft dataset for 21st June 2012. This is due to the fact that the AUC results for LOBSTER stocks (except for Microsoft) are not available from other models, so only Microsoft stock is reported. However, for some state-of-the-art models like AHMMAS [27],

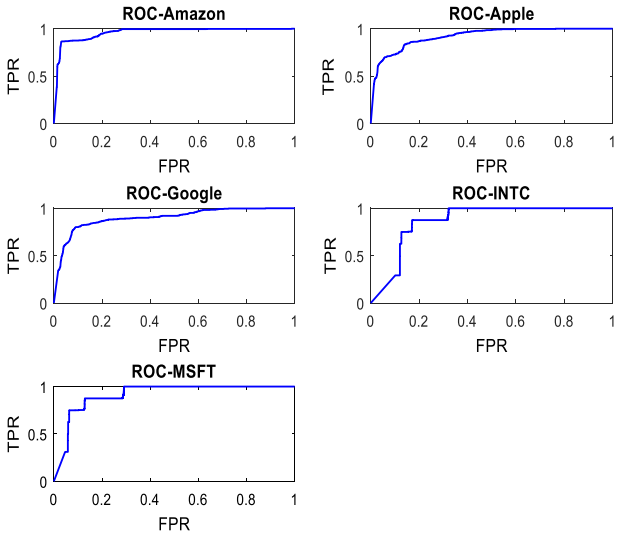

FIGURE 7. ROC curves for five different stocks. Group A (Amazon, Apple, Google) stocks show an identical behaviour in their performance that can be attributed to their smaller data size and the similar amount of anomalies injected whereas Group B (Microsoft, Intel Corp) stocks provide a different (almost similar performance within each other) compared to Group A attributing to the larger injection of anomalies into them.

where all the details about the parameters used for the same dataset (and using a combination of different anomalies), the proposed approach is again compared for the rest of the stocks in tables 4, 5 and 6. AUC Comparison for specific manipulation type with the existing state-of-the-art models is made impossible since most of the existing benchmark models have not provided results under specific manipulation type using same stocks and replicating their models is made impossible due to missing parameters values.

TABLE 2. Comparison of AUC for all five stocks when the manipulation occurs within close vicinity of each other.

	Anomalies placed far from each other **(1000ms apart)**	Anomalies placed close to each other **(6ms apart)**	% fall in AUCs
AAPL	0.9206	**0.8773**	4.70
AMZN	0.9602	**0.9539**	0.65
GOOG	0.8996	**0.8923**	0.81
INTC	0.8680	**0.7994**	7.90
MSFT	0.9143	**0.8804**	3.70

In order to check for the robustness of the proposed approach in detecting manipulations when two or more manipulative activities occur within a short duration of itself, the KPCA-MKDE based clustering model is applied on a dataset where the artificial anomalies are placed close to each other. Results are calculated after injecting same three anomalies described before, placed only 6 ms apart from each other. Table 2 shows a comparison of AUCs so calculated with the arrangement when they are separated 1000 ms apart on an average. It can be clearly seen from Table 2 that the

fall in AUC values for a situation when the anomalies are placed sufficiently close to each other is not more than 8%, (Intel Corp data [*Group B*]). For stocks like Amazon and Google [*Group A*], there is only a small fall in AUC as $<1\%$ change is encountered in both the stocks. The derived inference from such results is that although there is a vast change between the two situations in terms of spacing among different manipulation activities, the robustness of detection model remains intact.

TABLE 3. ***p*-value for the MKDE estimate for first seven principal components calculated.**

	PC_1	PC_2	PC_3	PC_4	PC_5	PC_6	PC_7
Amazon	1.22e-08	1.62e-07	0.0001	2.24e-16	3.34e-14	3.56e-41	1.28e-31
Apple	5.91e-18	2.05e-35	7.36e-06	6.62e-18	9.57e-07	4.016e-66	7.52e-09
Google	1.81e-05	8.37e-42	0.0523	0.04301	3.49e-14	1.052e-08	9.078e-11
INTC	4.71e-26	0	2.59e-43	0	0.0068	2.31e-06	0
MSFT	8.59e-11	0.0052	2.05e-93	0	7.49e-22	2.56e-40	0.0037

The class discrimination capability of the principal components from KPCA was assessed using Kruskal-Wallis statistical test as it fits for mutually independent components and avoids the assumption that the underlying datasets are inherently normally distributed [64]. Chi-square is used as a test statistic here to evaluate the performance of the proposed method. In table 3, the *p*-values for every individual principal component obtained from KPCA for both normal and manipulative trading instances in *group I* stocks are presented. Smaller *p*-values (less than 0.05) obtained for every principal component proves the statistical significance of the proposed model using KPCA. However, since the significance levels are variable among all the components, manipulation detection is not possible by defining a single threshold. The detection ability of the proposed approach between normal and abnormal classes is further evaluated using the following performance metrics: AUC, FAR [23], [27], [28], [57], [62], [63] and F-measure [23], [27], [28], [59]. The corresponding values for AUC, F-measure and FAR are summarised and compared with the existing approaches in Table 4-6 respectively.

TABLE 4. **AUC comparison of KPCA-MKDE approach with benchmark techniques for anomaly detection.**

	KPCA-MKDE	kNN [67]	PCA [45]	k-means [66]	OCSVM [67]	AHMMAS [27]
Amazon	**0.9602**	0.7982	0.9013	0.5799	0.8933	0.5152
Apple	**0.9206**	0.7926	0.6902	0.5819	0.6603	0.5344
Google	**0.8996**	0.5612	0.7993	0.6328	0.5911	0.5119
INTC	**0.8732**	0.5469	0.868	0.5077	0.697	0.5169
MSFT	**0.9143**	0.5509	0.8655	0.5047	0.6419	0.6711

Furthermore, the proposed detection model is repeatedly applied over *group I* dataset by varying window sizes to

TABLE 5. **F-measure comparison of KPCA-MKDE approach with benchmark techniques for anomaly detection.**

	KPCA-MKDE	kNN [67]	PCA [45]	k-means [66]	OCSVM [67]	AHMMAS [27]
Amazon	**0.5559**	0.1714	0.1568	0.0484	0.0284	0.0102
Apple	**0.6394**	0.0344	0.0457	0.0708	0.0045	0.0012
Google	**0.5651**	0.135	0.0806	0.0513	0.0196	0.0072
INTC	**0.6034**	0.1014	00085	0.0119	0.0126	0.0175
MSFT	**0.6216**	0.1148	0.0077	0.0141	0.0092	0.0279

TABLE 6. **FAR comparison of KPCA-MKDE approach with benchmark techniques for anomaly detection.**

	KPCA-MKDE	kNN [67]	PCA [45]	k-means [66]	OCSVM [67]	AHMMAS [27]
Amazon	1.22	**0.14**	3.9	7.33	49.54	9.22
Apple	1.07	**0.45**	6.64	1.26	67.8	7.83
Google	1.62	0.68	7.22	9.95	75.2	**0.5**
INTC	0.54	0.23	57.29	**0.02**	59.08	1.15
MSFT	0.71	0.08	49.89	**0.02**	77.48	0.52

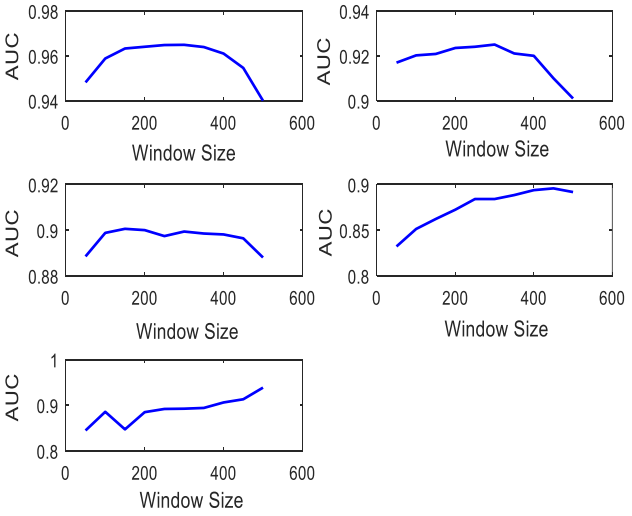

FIGURE 8. **Varying AUC values with number of samples fed to multi-dimensional KDE clustering algorithm. Optimal window length for Amazon, Apple and Google can be observed around 300 sample. Intel and Microsoft's AUCs rises with increasing number of samples.**

MKDE based clustering. It is performed to reduce the amount of uncertainty over the number of samples to be used as an input to clustering. The evaluation assessment in such a case is again carried out using AUC as a performance measure. Figure 8 shows the variability of AUCs with different window sizes. It can be easily inferred from this Figure that the AUC values for stocks: Amazon, Apple and Google rise with window sizes initially but falls when the number of samples exceeds a given value (300 samples /window). For Intel and Microsoft stocks, the AUC value continues to increase and is maximum when window size is 500. The average spacing among anomalies in this case is 1000 msec.

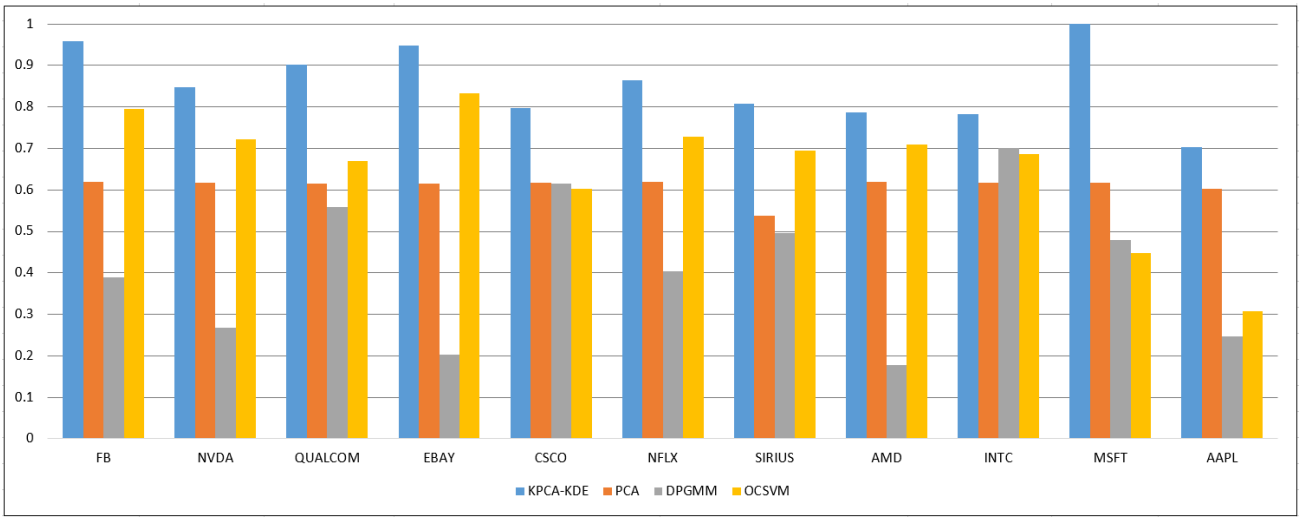

FIGURE 9. F-measure comparison for KPCA-MKDE approach on NBS dataset with existing benchmark anomaly detection approaches.

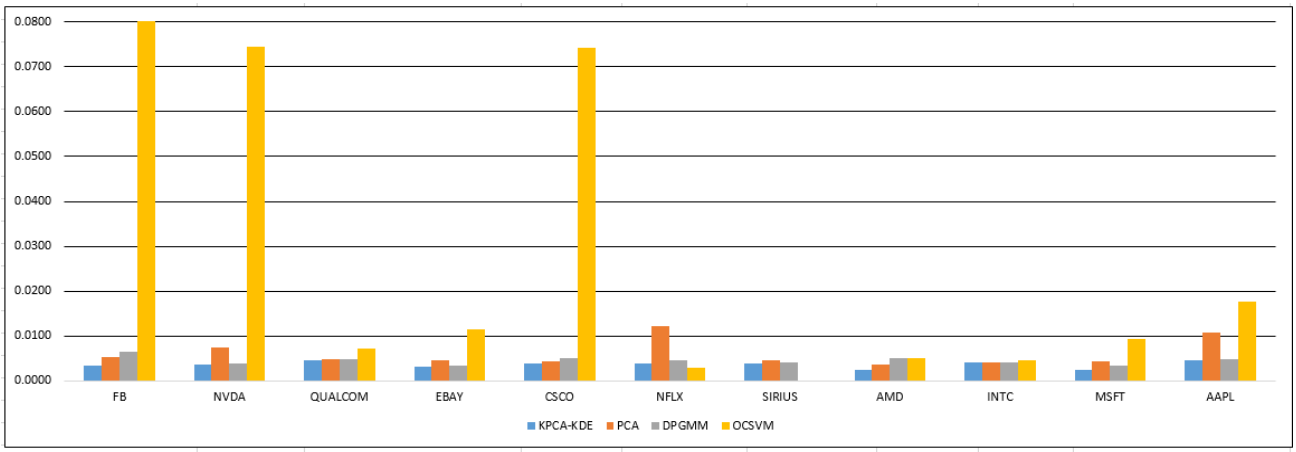

FIGURE 10. FAR comparison for KPCA-MKDE approach on NBS dataset with existing benchmark anomaly detection approaches.

A more exhaustive evaluation of the proposed approach is made by including other performance measures like AUC, F-measure and false alarm rate for the same dataset. Table 4, 5 and 6 mentions a comparative analysis of the KPCA-MKDE based approach using such measures. It can be easily interpreted from the tables that though the AUCs and F-measures for the proposed approach surpassed the existing anomaly detection approaches in unsupervised learning, there are some downsides when it comes to false positives. As mentioned in Table 6, although some of the existing approaches have better FAR values than the proposed approach, the overall performance can still be appreciated as it provided significant improvement in terms of F-measures and AUC values.

The proposed approach is also applied on *group II* dataset taken from Bloomberg trading platform. A more recent dataset of 11 stocks from 12th November 2018 is also considered from Bloomberg Trading platform in Northumbria Business School (NBS). The stocks considered here are selected because of their popularity and high trading frequency and the total number of trades (Average number of trades per stock per day $\sim$ 1 million). Figure 9 and 10 shows the F-measure and the False Alarm Rates (FAR) also called as FPR obtained using the proposed model. As it can be observed, the proposed model proves to be efficient and clearly outperforms the existing models for stock price manipulation detection.

The experimental results obtained using KPCA – MKDE based approach achieves a higher rate of detection of manipulation of three types (Saw tooth, Spike & Square pattern) in stock price. Manipulation schemes like pump and dump, ramping and quote stuffing are carefully modelled by the time series following real life cases reported by SEC [15], [17], [18]. The results also outperformed some of the existing approaches for stock price manipulation detection and also some of the existing benchmark techniques for anomaly detection like PCA [45], K-means [66], kNN [67],

OCSVM [67] and AHMMAS [27]. Such a performance can be attributed to the wider information content revealed due to the adaptation of the principal components from KPCA by increasing the spread of the data points. Such a spread is later exploited by MKDE to cluster normal trades. The robustness of the proposed approach can be explained from the decomposition of the feature sets for a given length of the samples in a window using KPCA. In reference to the cases of price manipulation for pump and dump and quote stuffing, a sudden flip in prices after a long held position of incremental rise (within the selected window of data samples) in prices arouses an uncertainty over the sample length for clustering of the dataset. To further explore such an issue, variable window sizes were considered during the experiment with stocks and the results so calculated. While for *Group A* stocks: Amazon, Apple and Google, AUC achieved maximum attainment around an optimal window length of 300 samples per window for MKDE based clustering approach, *Group B* stocks: Amazon and Intel Corp. on the other hand continues to rise even if the window size is increased up to 500. Although, the research cannot contribute in explaining the possible rationale behind such variations in AUCs, further investigation into such a behaviour of the model reveals that Intel and Microsoft stock prices usually sustain a given value (piecewise constant) for a considerable amount of time rather than frequent variation as in *Group A* stocks. Figure 6 shows such a behaviour during a same period from 09:30:00 AM to 09:30:52 AM for all the stocks prices. The robustness of the KPCA-MKDE approach is capable to achieve higher detection rates even when several manipulation schemes occurs successively. Only a small change ($<1\%$ fall) in AUCs is observed for Amazon and Google stocks when the anomalies are placed close to each other (6 ms apart) as compared to when they are sufficiently far apart. Even the least AUC value achieved for Intel data (0.7992) in the former case is still close 0.8, which is considered better performance for a classifier [65]. The proposed model for manipulation detection performed variably for some of the data sets and did not attain very high values of AUC as it did for Amazon, Apple and Microsoft stocks. This can be attributed to the high variability of the data and the possible overlapping of the anomalies with similar waveforms that created large False Positives (FPs), nevertheless still managed to get AUC values higher than the rest of the existing approaches. Apart from the AUC results, the values from Table 5 and 6 elaborates the detection outcomes. It is observed that the F-measure values for the proposed approach are not very significant (although comparatively) in values (<0.65). Further investigation into such an issue reveals the degraded detection performance of the approach towards spoofing manipulation schemes. This is due to the drawback of the level-1 tick data being used, as it does not contain the order cancellation information. This is crucially informative as it correlates the price fluctuation (usually assumed high for spoof trading) with the volume change. This information can be included in a future investigation using level-2 order book implying the price volatility associated with the order cancellation may lead to improved F-measure and false alarm rates.

To test the validity and robustness of the proposed algorithm, it is further tested on a recent dataset acquired from Bloomberg trading platform, NBS having 11 different stocks. The F-measure and FAR values generated are shown in Figure 9 and 10. It can easily be interpreted from the Figures that the proposed approach outperforms the benchmark anomaly detection techniques like PCA, OCSVM and DPGMM. The major contribution to such a performance is attributed to the ability of the multidimensional KPCA-MKDE algorithm to distinguish between normal and manipulative trades and to the less volatile nature of the stocks included. However, the FAR value, figure 10 for Netflix (NFLEX) and SIRIUS stocks is degraded compared to OCSVM but is accompanied with a considerable compensation for the same stocks in terms of F-measure, as can be seen from figure 9. It is worth mentioning here that the computational complexity of such an adept approach is $O(m^3)$ to decompose the m-dimensional input data using KPCA using RBF kernel. Upon proper selection of the principal components, the total number of dimensions of the KPCA output have been reduced to l dimensions. Furthermore, it requires only $O(N.l.\log(N) + 2^j)$ calculations for clustering using Multivariate KDE via diffusion [48] with 'N' samples in a given window, l variables and j clusters.

VII. CONCLUSION

This paper presented an innovative approach for detecting stock price manipulation based on the combination of KPCA and MKDE based clustering. A brief review of the literature covering detection of market abuses has also been presented along with their limitations. This research proposed to use an unsupervised learning model for detecting stock price manipulation. To test the validity of the proposed model, two real world stock datasets comprising of 16 different datasets (13 stocks in total) were used and augmented using artificially generated manipulation cases. Principal components were computed, through a non-linear transformation using the kernel trick, upon a set of features extracted from the stock prices. The dataset is then time-windowed before passing the selected components to the MKDE clustering algorithm for manipulation detection. The MKDE clustering algorithm groups the multivariate input dataset into clusters based on the density estimate defined within a bandwidth parameter. A threshold value set up on the clustered region separates the normal and anomalous trading instances. Different performance metrics such as AUC, F-measure and FAR were used to evaluate the performance of the proposed approach. A comparative analysis of the proposed approach results is performed with existing price manipulation detection researches and also with existing unsupervised anomaly detection techniques.

It can be easily observed that the proposed model outperformed existing manipulation detection techniques in terms of improving the AUC, enhancing the F-measure and

reducing the false alarm rates while totally avoiding the labelling information. Such an improvement in the results was leveraged from the non-linear decomposition of stock prices using KPCA and further adaptation of the decomposed components. This helped in increasing the gap between the normal and abnormal stock trades in the transformed kernel domain. For further research, the performance of the proposed approach can be evaluated by varying the kernel functions for both KPCA and MKDE. In addition, the inclusion of the volume information for the cancelled orders using level-2 data will be considered for further enhancement of the detection performance.

REFERENCES

[1] *Financial Conduct Authority: MAR 1.6 Market Abuse*, Financial Conduct Authority, London, U.K., no. 5, 2014, sec. 118.
[2] F. Allen and D. Gale, "Stock-price manipulation," *Rev. Financial Stud.*, vol. 5, no. 3, pp. 503–529, 1992.
[3] M. J. Aitken, F. H. Harris, and S. Ji, "Trade-based manipulation and market efficiency: A cross-market comparison," in *Proc. 22nd Australas. Finance Banking Conf.*, Sydney, NSW, Australia, 2009, pp. 1–43.
[4] C. Pirrong, "The economics of commodity market manipulation: A survey," *J. Commodity Markets*, vol. 5, no. 3, pp. 1–17, Mar. 2017.
[5] S. Neupane, S. G. Rhee, K. Vithanage, and M. Veeraraghavan, "Trade-based manipulation: Beyond the prosecuted cases," *J. Corporate Finance*, vol. 42, pp. 115–130, Feb. 2017.
[6] E. J. Lee, K. S. Eom, and K. S. Park, "Microstructure-based manipulation: Strategic behavior and performance of spoofing traders," *J. Financial Markets*, vol. 16, no. 2, pp. 227–252, May 2013.
[7] D.-Y. Yeung and Y. Ding, "Host-based intrusion detection using dynamic and static behavioral models," *Pattern Recognit.*, vol. 36, no. 1, pp. 229–243, Jan. 2003.
[8] X. Ding, Y. Li, A. Belatreche, and L. P. Maguire, "An experimental evaluation of novelty detection methods," *Neurocomputing*, vol. 135, pp. 313–327, Jul. 2014.
[9] H. Hoffmann, "Kernel PCA for novelty detection," *Pattern Recognit.*, vol. 40, no. 3, pp. 863–874, Mar. 2007.
[10] S. Alexander, O. Mangasarian, and B. Schölkopf, "Sparse kernel feature analysis," Data Mining Inst., Max-Planck-Gesellschaft, Munich, Germany, Tech. Rep. 99-04, 1999.
[11] L. C. Matioli, S. R. Santos, M. Kleina, and E. A. Leite, "A new algorithm for clustering based on kernel density estimation," *J. Appl. Statist.*, vol. 44, no. 1, pp. 1–20, 2016.
[12] T. C. W. Lin, "The new market manipulation," *Emory Law J.*, vol. 66, p. 1253, Jul. 2017.
[13] FINRA, Washington, DC, USA. (2012). *Finra Joins Exchanges and the SEC in Fining Hold Brothers More Than $5.9 Million for Manipulative Trading, Anti-Money Laundering, and Other Violations*. [Online]. Available: http://www.finra.org/Newsroom/NewsReleases/2012/P178687
[14] D. Diaz and B. Theodoulidis, "Financial markets monitoring and surveillance: A quote stuffing case study," Social Sci. Res. Netw. (SSRN), Amsterdam, Netherlands, Tech. Rep. 2913636, 2012.
[15] *SEC vs ATG Capital LLC*, Securities Exchange Commission, New York, NY, USA, Mar. 2019.
[16] N. Hautsch and R. Huang, "The market impact of a limit order," *J. Econ. Dyn. Control*, vol. 36, no. 4, pp. 501–522, 2012.
[17] Nanex, Denver, CO, USA. (2012). *Whac-a-Mole is Manipulation*. [Online]. Available: http://www.nanex.net/aqck2/3598.html
[18] *SEC vs Lidingo Holdings LLC*, Securities Exchange Commission, New York, NY, USA, Apr. 2017.
[19] Y. S. Abu-Mostafa, A. F. Atiya, M. Magdon-Ismail, and H. White, "Introduction to the special issue on neural networks in financial engineering," *IEEE Trans. Neural Netw.*, vol. 12, no. 4, pp. 653–656, Jul. 2001.
[20] I. Domowitz, "Market abuse and surveillance," Foresight, Government Office Sci., London, U.K., Econ. Impact Assessment EIA17, 2012.
[21] S. Friederich and R. Payne, "Computer-based trading and market abuse," Foresight, Government Office Sci., London, U.K., Driver Rev. DR20, 2012.
[22] F. Yang, H. Yang, and M. Yang, "Discrimination of China's stock price manipulation based on primary component analysis," in *Proc. Int. Conf. Behav., Econ., Socio-Cultural Comput. (BESC)*, Shanghai, China, Oct. 2014, pp. 1–5.
[23] K. Golmohammadi, O. R. Zaiane, and D. Díaz, "Detecting stock market manipulation using supervised learning algorithms," in *Proc. Int. Conf. Data Sci. Adv. Anal. (DSAA)*, Shanghai, China, Oct. 2014, pp. 435–441.
[24] H. Öğüt, M. M. Doğanay, and R. Aktaş, "Detecting stock-price manipulation in an emerging market: The case of turkey," *Expert Syst. Appl.*, vol. 36, no. 9, pp. 11944–11949, Nov. 2009.
[25] T. Leangarun, P. Tangamchit, and S. Thajchayapong, "Stock price manipulation detection using a computational neural network model," in *Proc. 8th Int. Conf. Adv. Comput. Intell. (ICACI)*, Chiang Mai, Thailand, Feb. 2016, pp. 337–341.
[26] LOBSTER Project, Atlanta, GA, USA. (2012). *Limit Order Book System*. [Online]. Available: https://lobsterdata.com/info/DataSamples.php
[27] Y. Cao, Y. Li, S. Coleman, A. Belatreche, and T. M. McGinnity, "Adaptive hidden Markov model with anomaly states for price manipulation detection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 2, pp. 318–330, Feb. 2015.
[28] D. Diaz, B. Theodoulidis, and P. Sampaio, "Analysis of stock market manipulations using knowledge discovery techniques applied to intraday trade prices," *Expert Syst. Appl.*, vol. 38, no. 10, pp. 12757–12771, Sep. 2011.
[29] Z. Ferdousi and A. Maeda, "Unsupervised outlier detection in time series data," in *Proc. 22nd Int. Conf. Data Eng. Workshops (ICDEW)*, 2006, pp. 51–56.
[30] Y. Kim and S. Y. Sohn, "Stock fraud detection using peer group analysis," *Expert Syst. Appl.*, vol. 39, no. 10, pp. 8986–8992, Aug. 2012.
[31] Q. Wang, W. Xu, X. Huang, and K. Yang, "Enhancing intraday stock price manipulation detection by leveraging recurrent neural networks with ensemble learning," *Neurocomputing*, vol. 347, pp. 46–58, Jun. 2019.
[32] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 1–58, Jul. 2009.
[33] F. Allen and G. Gorton, "Stock price manipulation, market microstructure and asymmetric information," Nat. Bur. Econ. Res., Cambridge, MA, USA, Tech. Rep. 3862, 1991.
[34] D. Giannone, M. Lenza, and G. Primiceri, "Economic predictions with big data: The illusion of sparsity," Federal Reserve Bank New York, New York, NY, USA, Working Paper 847, 2017.
[35] Y.-K. Kwok, *Mathematical Models of Financial Derivatives*. Singapore: Springer, 1998.
[36] E. Haven, X. Liu, and L. Shen, "De-noising option prices with the wavelet method," *Eur. J. Oper. Res.*, vol. 222, no. 1, pp. 104–112, Oct. 2012.
[37] D. Donoho and I. M. Johnstone, "Ideal spatial adaptation by wavelet shrinkage," Dept. Statist., Stanford Univ., Stanford, CA, USA, Tech. Rep., 1993, vol. 81.
[38] B. Rizvi, A. Belatreche, and A. Bouridane, "A dendritic cell immune system inspired approach for stock market manipulation detection," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Wellington, New Zealand, Jun. 2019, pp. 3325–3332.
[39] I. Jolliffe, *Principal Component Analysis*. Hoboken, NJ, USA: Wiley, 2002.
[40] G. C. Ranger and F. B. Alt, "Choosing principal components for multivariate statistical process control," *Commun. Statist.-Theory Methods*, vol. 25, no. 5, pp. 909–922, Jan. 1996.
[41] T. Kourti and J. F. MacGregor, "Process analysis, monitoring and diagnosis, using multivariate projection methods," *Chemometric Intell. Lab. Syst.*, vol. 28, no. 1, pp. 3–21, Apr. 1995.
[42] Q. Wang, "Kernel principal component analysis and its applications in face recognition and active shape models," Rensselaer Polytech. Inst., Troy, NY, USA, Tech. Rep. 1207.3538, 2012. [Online]. Available: http://arxiv.org/ abs/1207.3538
[43] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Comput.*, vol. 10, no. 5, pp. 1299–1319, Jul. 1998.
[44] K. Heafield, "Detecting network anomalies with kernel principal component analysis," Netlab, California Inst. Technol. (Caltech), Pasadena, CA, USA, Res. Rep., 2006. [Online]. Available: http://kheafield.com/professional/netlab/final.pdf
[45] M. L. Shyu, S. C. Chen, K. Sarinnapakorn, and L. W. Chang, "A novel anomaly detection scheme using principal component classifier," in *Proc. 3rd IEEE Int. Conf. Data Mining*, Jan. 2003, pp. 353–365.
[46] W. Li, H. H. Yue, S. Valle-Cervantes, and S. J. Qin, "Recursive PCA for adaptive process monitoring," *J. Process Control*, vol. 10, no. 5, pp. 471–486, Oct. 2000.

[47] J. Ni, C. Zhang, and S. X. Yang, "An adaptive approach based on KPCA and SVM for real-time fault diagnosis of HVCBs," *IEEE Trans. Power Del.*, vol. 26, no. 3, pp. 1960–1971, Jul. 2011.
[48] Z. I. Botev, J. F. Grotowski, and D. P. Kroese, "Kernel density estimation via diffusion," *Ann. Statist.*, vol. 38, no. 5, pp. 2916–2957, Oct. 2010.
[49] M. Thomas, K. D. Brabanter, and B. D. Moor, "New bandwidth selection criterion for kernel PCA: Approach to dimensionality reduction and classification problems," *BMC Bioinf.*, vol. 15, no. 1, p. 137, Dec. 2014.
[50] G. R. Terrell and D. W. Scott, "Variable kernel density estimation," *Ann. Statist.*, vol. 20, no. 3, pp. 1236–1265, Sep. 1992.
[51] M. C. Jones and D. F. Signorini, "A comparison of higher-order bias kernel density estimators," *J. Amer. Stat. Assoc.*, vol. 92, no. 439, pp. 1063–1073, Sep. 1997.
[52] J. S. Marron and M. P. Wand, "Exact mean integrated error," *Ann. Statist.*, vol. 20, no. 2, pp. 712–736, 1992.
[53] M. P. Wand and M. C. Jones, "Comparison of smoothing parameterizations in bivariate kernel density estimation," *J. Amer. Stat. Assoc.*, vol. 88, no. 422, pp. 520–528, Jun. 1993.
[54] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*. London, U.K.: Chapman & Hall, 1986.
[55] B. Maillet and P. Merlin. (2009). *Outliers Detection, Correction of Financial Time-Series Anomalies and Distributional Timing for Robust Efficient Higher-Order Moment Asset Allocations*. [Online]. Available: http://ssrn.com/abstract=1413623
[56] L. Cao, Y. Ou, and P. S. Yu, "Coupled behavior analysis with applications," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 8, pp. 1378–1392, Aug. 2012.
[57] R. A. Craig and L. Liao, "Phylogenetic tree information aids supervised learning for predicting protein-protein interaction based on distance matrices," *BMC Bioinf.*, vol. 8, no. 1, pp. 1–12, Dec. 2007.
[58] S. Wang, D. Li, N. Petrick, B. Sahiner, M. G. Linguraru, and R. M. Summers, "Optimizing area under the ROC curve using semi-supervised learning," *Pattern Recognit.*, vol. 48, no. 1, pp. 276–287, Jan. 2015.
[59] P. F. Evangelista, M. J. Embrechts, P. Bonissone, and B. K. Szymanski, "Fuzzy ROC curves for unsupervised nonparametric ensemble techniques," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, Montreal, QC, Canada, Jul./Aug. 2005, pp. 3040–3045.
[60] R. Kharghanian, A. Peiravi, and F. Moradi, "Pain detection from facial images using unsupervised feature learning approach," in *Proc. 38th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Orlando, FL, USA, Aug. 2016, pp. 419–422.
[61] A. G. Roy and D. Sheet, "DASA: Domain adaptation in stacked autoencoders using systematic dropout," in *Proc. 3rd IAPR Asian Conf. Pattern Recognit. (ACPR)*, Kuala Lumpur, Malaysia, Nov. 2015, pp. 735–739.
[62] J. Wu and X.-L. Zhang, "Maximum margin clustering based statistical VAD with multiple observation compound feature," *IEEE Signal Process. Lett.*, vol. 18, no. 5, pp. 283–286, May 2011.
[63] I. Smal, M. Loog, W. Niessen, and E. Meijering, "Quantitative comparison of spot detection methods in fluorescence microscopy," *IEEE Trans. Med. Imag.*, vol. 29, no. 2, pp. 282–301, Feb. 2010.
[64] J. H. McDonald, *Handbook of Biological Statistics*. Baltimore, MD, USA: Sparky House, 2014.
[65] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, Jun. 2006.
[66] S. Chawla and A. Gionis, "K-means-: A unified approach to clustering and outlier detection," in *Proc. SIAM, SDM*, 2013, pp. 187–197.
[67] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.

BAQAR A RIZVI (Member, IEEE) received the M.Tech. degree (Hons.) in communication and information systems from Aligarh Muslim University, India, in 2013. He is currently pursuing the Ph.D. degree in stock price manipulation detection using data mining techniques with Northumbria University, Newcastle. He was a Research Associate with the Indian Institute of Technology, New Delhi, in 2014. He has published few peer-reviewed articles in these fields. His research interests include machine/deep learning for stock market data analysis, including stock volume data, bio-medical signal processing for EEG, ECG, and EMG signals, and bio-inspired optimization techniques.

AMMAR BELATRECHE (Member, IEEE) received the Ph.D. degree in computer science from Ulster University, Derry, U.K. He joined Northumbria University, in May 2016. He was a Research Associate with the Intelligent Systems Research Centre (ISRC) and a Lecturer in computer science with the School of Computing and Intelligent Systems, Ulster University. He is currently an Associate Professor in computer science and a Program Leader with the M.Sc. Advanced Computer Science, Department of Computer and Information Sciences. He is also a member of the Computational Intelligence and Visual Computing (CIVC) Research Group. He has extensive experience across academic and research and development. He has led a number of research and consultancy projects. He has successfully supervised/co-supervised eight Ph.D. students to completion. His research interests include machine learning and AI, bio-inspired intelligent systems, structured and unstructured data analytics, capital markets engineering, and image processing and understanding. He is a Fellow of the Higher Education Academy. He was a program committee member of several international conferences and journals. He serves as an Associate Editor for *Neurocomputing* (Elsevier). He served as a reviewer for several international conferences and journals.

AHMED BOURIDANE (Senior Member, IEEE) received the Ingenieur d'Etat degree in electronics from the Ecole Nationale Polytechnique of Algiers (ENPA), Algeria, in 1982, the M.Phil. degree in electrical engineering (VLSI design for signal processing) from Newcastle University, Newcastle upon Tyne, U.K., in 1988, and the Ph.D. degree in electrical engineering (computer vision) from the University of Nottingham, U.K., in 1992. From 1992 to 1994, he was a Research Developer in telesurveillance and access control applications. In 1994, he joined Queen's University Belfast, Belfast, U.K., as a Lecturer in computer architecture and image processing, where he was a Reader in computer science. He is currently a Full Professor in image engineering and security and Leads the Computational Intelligence and Visual Computing Group, Northumbria University, Newcastle. He has authored or coauthored more than 350 publications and two research books on *Imaging for Forensics and Security* and *Biometric Security and Privacy*. His research interests include imaging for forensics and security, biometrics, homeland security, image/video watermarking, medical engineering, cryptography, and mobile and visual computing.

IAN WATSON has 20 years of teaching experience in computer science and IT related programs. He has initiated and managed a number of International undergraduate and postgraduate programs, Far East. He is currently a Senior Lecturer and a Program Leader with Northumbria University.